

NGB_IN003_1 SZE 2014-15/2 (7)

Szoftver- minőségbiztosítás

Szoftvertesztelési folyamat

Szoftverek és környezet

- ❖ Nem egyforma a szoftverek használatához kapcsolódó kockázat
- ❖ **Különböző kockázati szintek -> eltérő tesztelési igény**
- ❖ Nem megfelelő szoftver viselkedés <- emberi eredetű **fejlesztési hibák** + a hiba manifesztálódásának feltételei
- ❖ Rendszer meghibásodások további okai (üzemel(tet)és közben):
 - ❖ **környezeti tényezők, emberi figyelmetlenség, helytelen használat, szándékos károkozás**

A tesztelés szerepe

- ❖ **Programhibák észlelése (fejlesztés, karbantartás)**
- ❖ **-> hiba előfordulás csökkentése üzemeltetési környezetben**
- ❖ **Felhasználói hibák esélyének csökkentése**
- ❖ **Termék és szolgáltatás minőség javítása**
- ❖ **Jogi feltételek (szabványok) betartása**

A tesztelés definíciója

- ❖ A tesztelés **folyamat** - tevékenységek sorozata
- ❖ Az összes szoftverfejlesztési életciklushoz kapcsolható
- ❖ Szoftver- és kapcsolódó termékek
- ❖ Többféle tevékenység
 - ❖ tervezés
 - ❖ előkészítés
 - ❖ végrehajtás
 - ❖ értékelés

Tesztelési célok

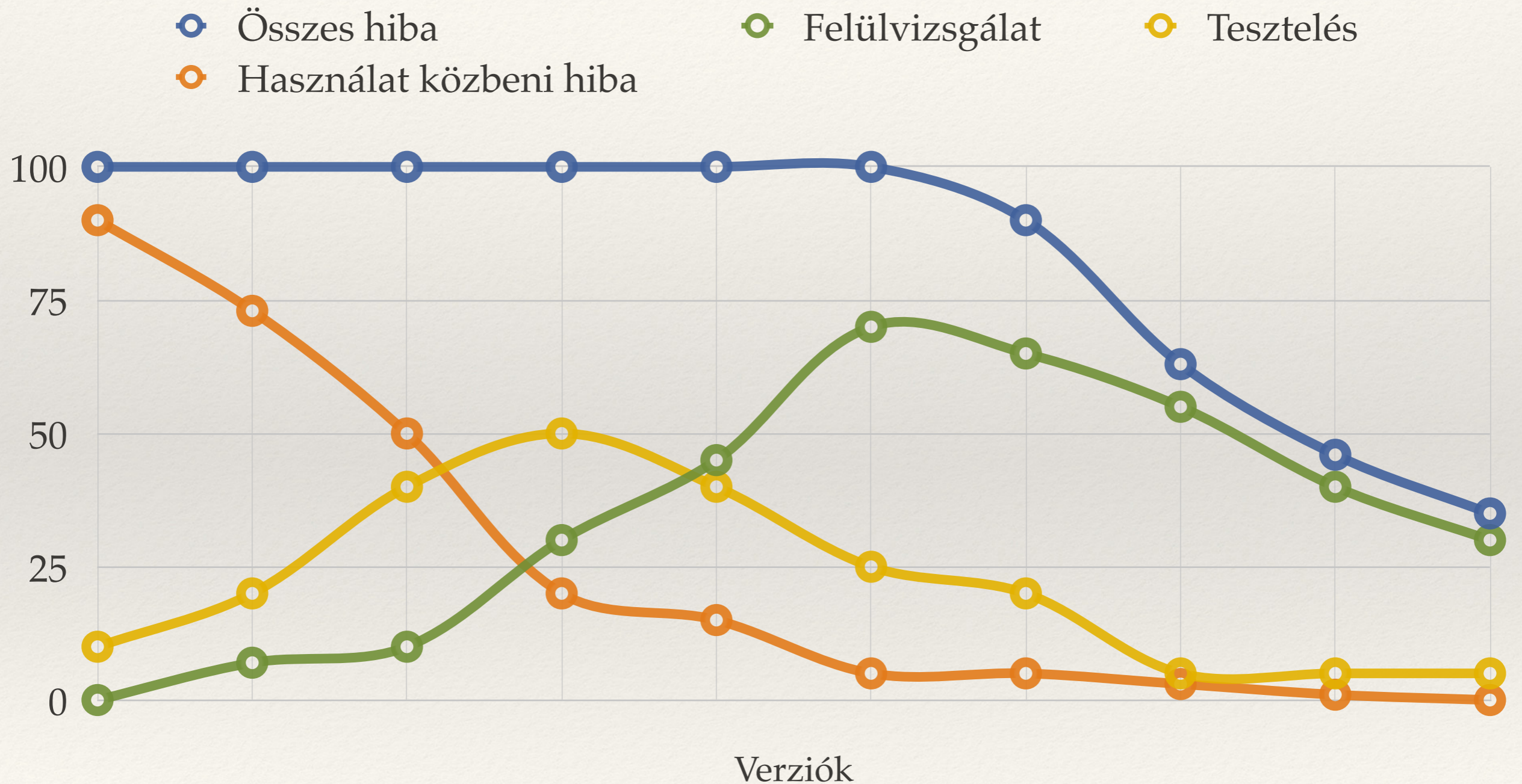
- ❖ Magas szintű cél: a rendszer és a fejlesztési folyamat javítása (információ)
- ❖ Közvetlen célok:
 - ❖ programhibák detektálása
 - ❖ minőség szintjének meghatározása
 - ❖ programhibák megelőzése



Hibakeresés és eltávolítás

- ❖ Teszt által detektált programhiba
 - ❖ **lokalizálás - hibakeresés**
 - ❖ javítás
 - ❖ ellenőrzés
- ❖ Hibakeresési ciklus: hibalokalizálás, javítás, újra tesztelés (fejlesztő)
- ❖ Regressziós tesztelés: javítás során nem került be hiba

Hiba eltávolítás (pl.)



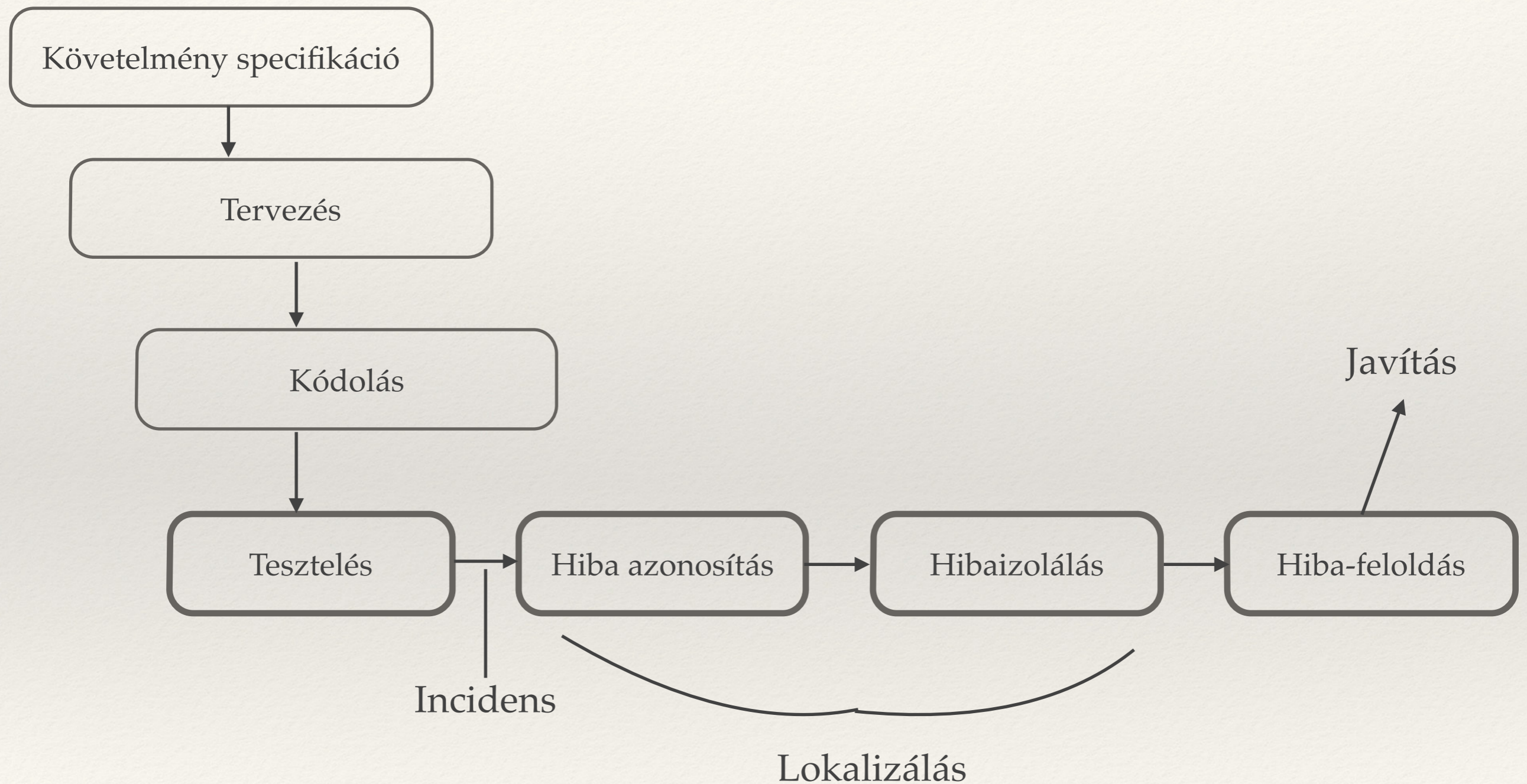
Tesztelési alapelvek

- ❖ A tesztelés fő célja **programhibák megtalálása**
- ❖ **Nem lehetséges kimerítő tesztelés** (bementek és előfeltételek minden kombinációjára)
- ❖ A tesztelést az életciklus lehető legkorábbi fázisában el kell kezdeni (gazdaságosság)
- ❖ Jellemző a programhibák csoportosulása
- ❖ A programhiba- csoportok idővel átalkulnak ("féregírtó paradoxon")
- ❖ A tesztelés függ a körülményektől (kockázatok)
- ❖ A gyakorlati használhatóság nem feltétlenül függ a programhibáktól ("a hibátlan rendszer téveszméje")

A tesztelési folyamat

- ❖ Főbb tevékenységek:
 - ❖ tervezés, irányítás
 - ❖ elemzés és műszaki teszttervezés
 - ❖ megvalósítás és végrehajtás
 - ❖ kilépési feltételek ellenőrzése és riportolás
 - ❖ tesztelés lezárása

A tesztelési életciklus



Teszttervezés

- ❖ Projekt célok és követelmények megértése
- ❖ Tesztelési megközelítés, **stratégia** meghatározása
 - ❖ tesztelési irányelvek (szervezeti vagy szoftver rendszer szintű alapelvek, általános szabályok)
- ❖ **Tesztterv** kidolgozása:
 - ❖ tesztelés hatókörének (scope) megállapítása
 - ❖ üzleti-, termék-, projekt-, technikai kockázatok
 - ❖ tesztelési megközelítés(ek) meghatározása
 - ❖ tesztelemelek, technikák, teamek, lefedettség, erőforrások
 - ❖ ütemterv
 - ❖ kilépési feltételek meghatározása

Teszt felügyeleti tevékenység

- ❖ Felülvizsgálatok és tesztelési eredmények elemzése (tesztek mennyisége, hibatalálási ráta, hibák súlyossága)
- ❖ Dokumentációs folyamat, információ visszacsatolás (fejlesztők, menedzsment, ügyfél, validátor / auditor)
- ❖ Javítások menedzselése
- ❖ Projekt döntéshozatal

Formális teszt monitorozás

- ❖ Célok:
- ❖ biztosítani, hogy a tesztelési eljárások összhangban vannak a tesztelési tervekkel
- ❖ biztosítani a tesztelési eljárások **verifikálhatóságát**
- ❖ biztosítani a teszt eljárások betartását
- ❖ biztosítani a tesztriportok pontosságát és teljességét
- ❖ biztosítani a tesztelés során **detektált hibák tényleges kijavítását**



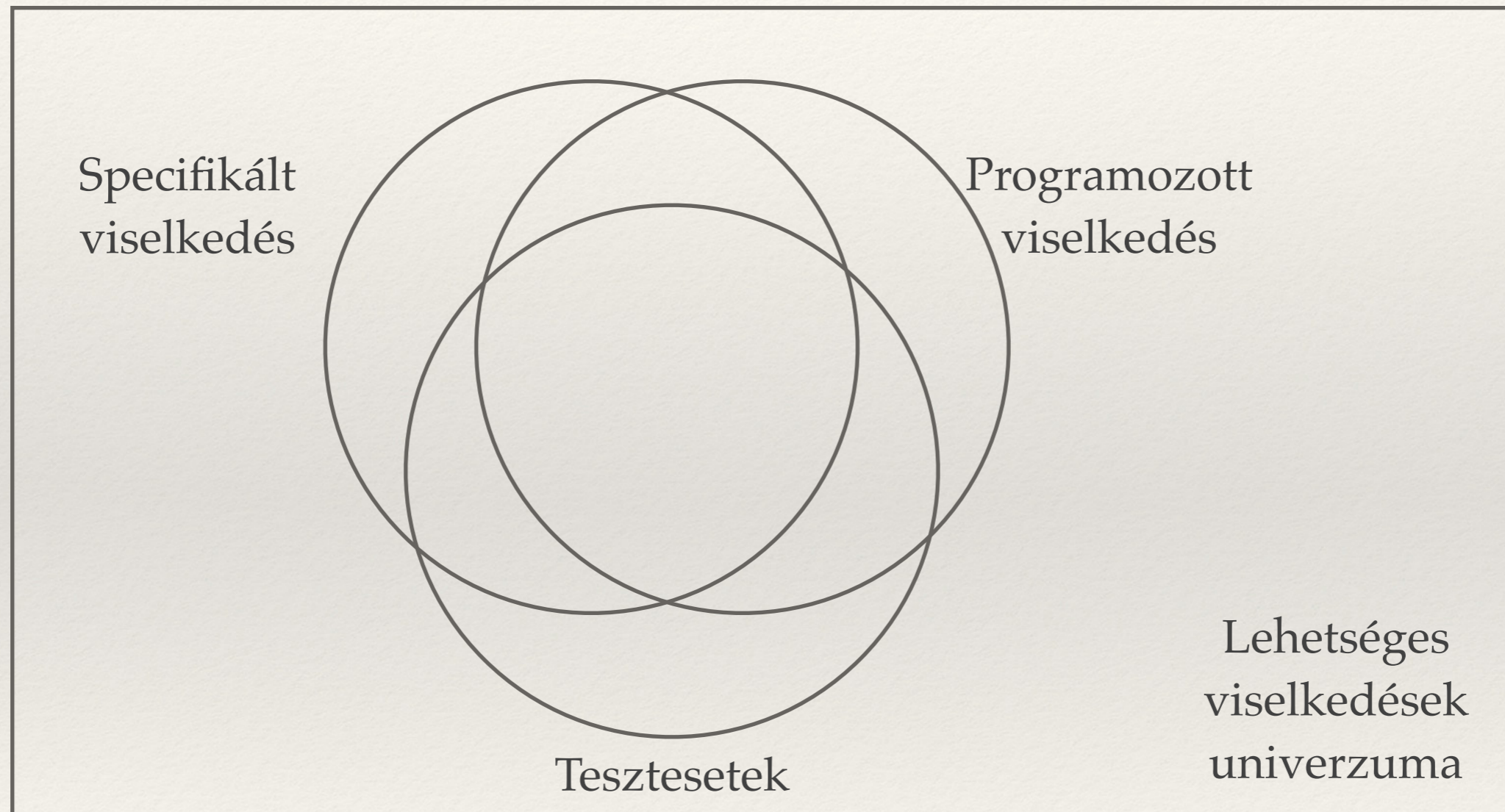
Műszaki tesztervezés

- ❖ általános tesztelési célok -> konkrét tesztervek, tesztelési feltételek
- ❖ tesztbázis kialakítása (dokumentumok, melyeken a tesztesetek alapulnak)
- ❖ specifikáció elemzése (javítás, pontosítás)
- ❖ tesztelési prioritások meghatározása
- ❖ tesztadatok meghatározása (statikus, dinamikus)
 - ❖ generálás (anonimizálás, szimuláció, stb.)
- ❖ tesztkörnyezet kialakítása (eszközök, infrastruktúra)

Tesztesetek

- ❖ **Azonosítóval** rendelkeznek
- ❖ Egy adott program-viselkedéshez kapcsolódnak
- ❖ Rögzített **előfeltételek** (precondition), kezdeti állapot (initial state)
- ❖ Meghatározott **inputok és outputok** (akciók és reakciók)
- ❖ Rögzített **végállapot** (post condition)

Tesztek és viselkedés



A tesztelés végrehajtása

- ❖ **Tesztesetek meghatározása, rögzítése**
- ❖ **Tesztelési eljárások kialakítása**
 - ❖ tesztszkriptek írása
 - ❖ futtatás automatizálása
 - ❖ megfelelő formátumú tesztadatok kialakítása
- ❖ **Tesztkészletek létrehozása**
- ❖ **Tesztkörnyezet megvalósítása, kialakítása**
- ❖ **Tesztek futtatása**
- ❖ **Eredmények naplózása**
- ❖ **Incidensek jelentése, változtatási igények rögzítése (->javítás)**

Független tesztelés

- ❖ Szoftver fejlesztés és tesztelés eltérő gondolkodásmódot igényel
- ❖ Különböző szintek:
 - ❖ fejlesztő
 - ❖ más személy a csapatból
 - ❖ más szervezeti egység tagja
 - ❖ más (külső) szervezet tagja
- ❖ Belső tesztelés előnyei
 - ❖ fejlesztési technológia, kód ismerete, hatékony hibalokalizálás
- ❖ Független tesztelés előnyei
 - ❖ elfogulatlanság, tesztelési professzionalizmus, hatékony hibadatektálás

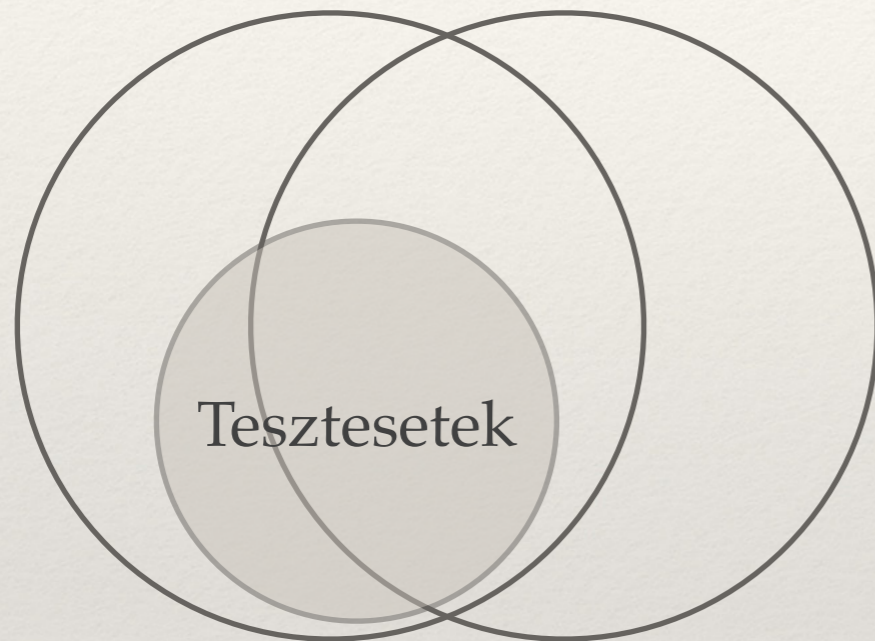
Tesztelési szintek

- ❖ Egység teszt (unit test)
 - ❖ legkisebb tesztelhető komponensek tesztje
- ❖ Modul teszt (module test)
 - ❖ egymással összefüggő komponens csoport tesztje
- ❖ Alrendszer teszt (subsystem test)
 - ❖ alrendszert alkotó modulok tesztje, interfésztesztek
- ❖ Rendszer teszt (systemtest, integration test)
 - ❖ integrált rendszer tesztje (funkc. és nem funk. köv., együttműködés, telj.)
- ❖ Átvételi teszt (acceptance test)
 - ❖ szolgáltatások tesztelése, valós viszonyok között

Tesztelési alap-megközelítések

- ❖ **Funkcionális** (black-box, specifikáció alapú) megközelítés
 - ❖ specifikációtól való eltérés felderítése a kód figyelembevétele nélkül
 - ❖ "bizalom" erősítése
- ❖ **Struktúrális** (white-box, lefedettség alapú) megközelítés
 - ❖ a kód minél alaposabb (utasítások, elágazások, utak) bejárása (végrehajtása) hibák felfedezésére
 - ❖ hibák keresése

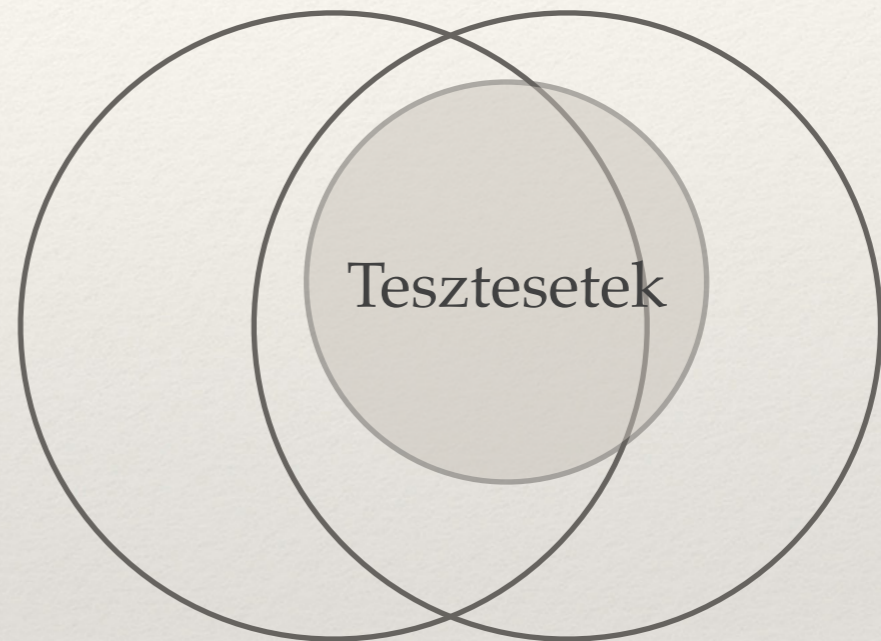
Tesztelési alap-megközelítések (folyt.)



Specifikált

Programozott

Funkcionális tesztek



Specifikált

Programozott

Struktúrális tesztek

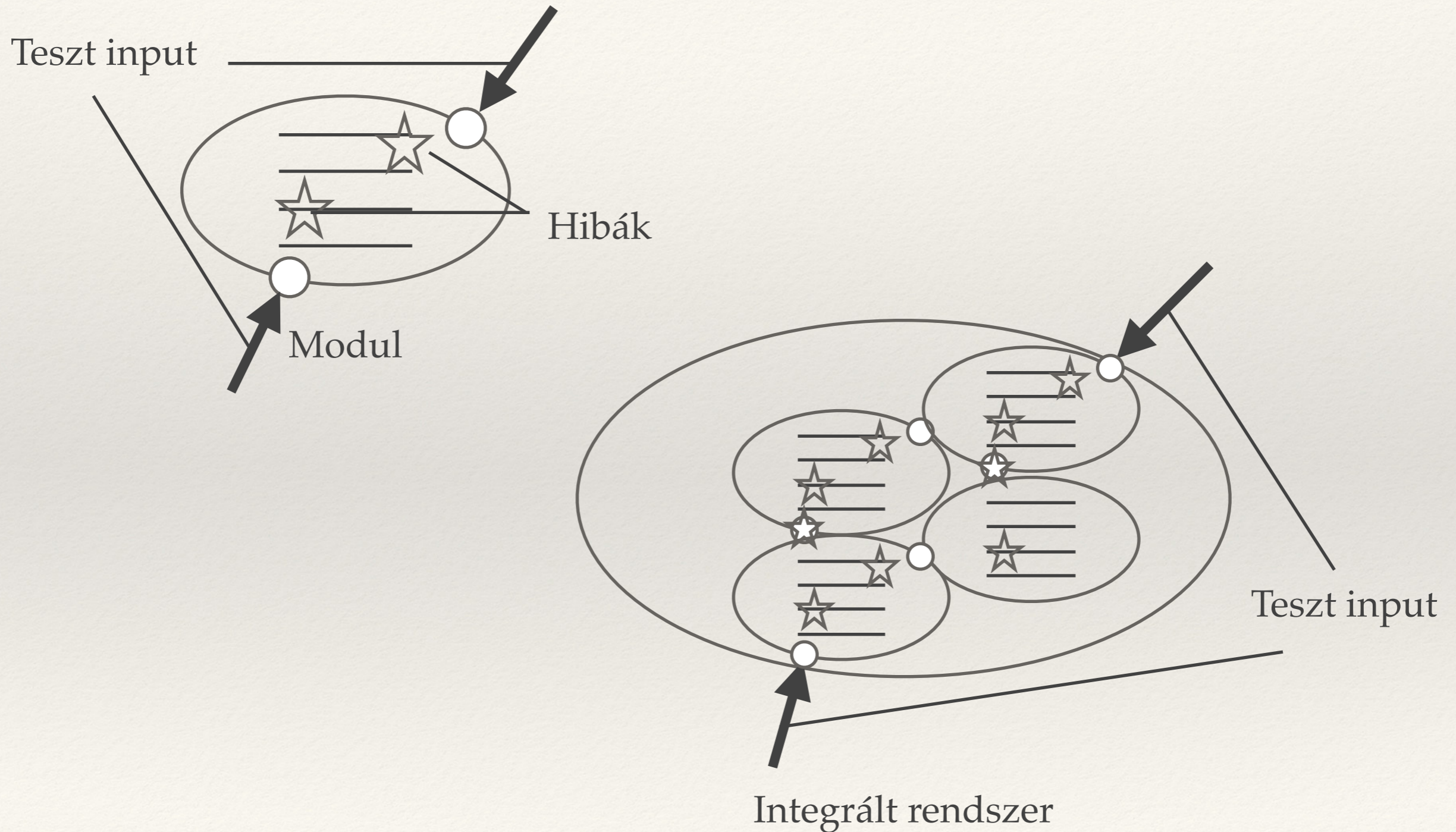
Egység- és modultesztek

- ❖ Moduláris kód (függvények, osztályok, metódusok) -> egységenkénti tesztelés
- ❖ kódolással szorosan összekapcsolva
- ❖ inkább white-box jellegű (+ specifikáció alapú tesztesetek)
- ❖ kódrészek elszigetelt (izoláció) végrehajtása
- ❖ korai tesztelhetőség
- ❖ felhasználás regressziós tesztelésre is
- ❖ modul interfész tesztelése
- ❖ jó idő kihasználás (hibák nagyrésze felderíthető, hatékony integrálás)
 - ❖ hatékony hibalokalizálás
 - ❖ rugalmas, jól kontrolálható input megvalósítás
 - ❖ könnyű és kényelmesebb létrehozás, karbantartás

Integrációs tesztelés

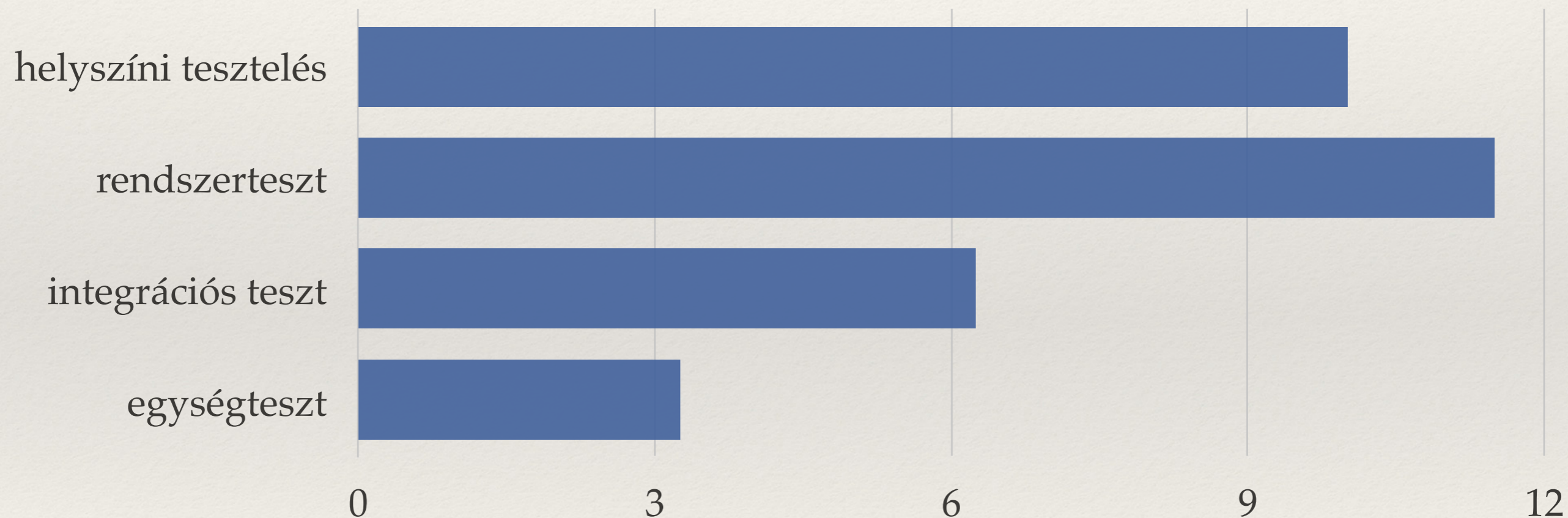
- ❖ Több "összeszerelt" komponens együttesének tesztelése
 - ❖ komponensek közötti kölcsönhatások tesztelése (együttműködés)
 - ❖ interfészek együttes tesztelése
- ❖ Megközelítések:
 - ❖ big bang
 - ❖ inkrementális (modulok hozzáadása lépésenként)

Izolációs és integrált modul tesztelés



Izolációs és integrált modul tesztelés (folyt.)

Különböző típusú tesztek egymáshoz viszonyított relatív időigénye



Magasabb szintű tesztek

- ❖ **Rendszerteszt**
 - ❖ követelmények -> specifikáció
 - ❖ rendszer céljai, felhaszn. doku. -> tesztesetek
 - ❖ független tesztelő, végfelhasználók (megrendelő)
- ❖ **Szolgáltatás teszt**
 - ❖ funkciók tesztelése nem a specifikációra építve
 - ❖ tevékenységek elvégezhető-e a szoftverrel
- ❖ **Mennyiségi és stresszteszt**
 - ❖ kapacitás korlátok ellenőrzése (nagy mennyiségű adat)
 - ❖ stressz teszt: erős (intenzív, szélsőséges) terhelés korlátozott idő alatt, robusztusság tesztelése
- ❖ **Használhatósági teszt**
 - ❖ használat közbeni minőség (hatékony és elégedettségre vezető használat)
 - ❖ felhasználói felület

Magasabb szintű tesztek (folyt.)

- ❖ Biztonsági teszt
 - ❖ adatbiztonsággal, adatvédelemmel kapcsolatos hibák (pl. pentest)
 - ❖ jogosultságok, adathelyesség, integritás
- ❖ **Teljesítményteszt**
 - ❖ specifikált teljesítménytől való eltérés felderítése
 - ❖ válaszidők, feldolgozási sebesség
- ❖ **Konfiguráció-teszt**
 - ❖ eltérő környezetekben való működés (HW, SW környezetek)
 - ❖ minimum és maximum konfiguráció
 - ❖ kompatibilitási tesztek
- ❖ Megbízhatósági teszt
 - ❖ meghibásodási ráta (sztochasztikus becslések)
 - ❖ visszaállási (recoverability) képesség tesztelése

Változtatásokhoz kapcsolódó tesztelés

- ❖ **Ellenőrző újratesztelés (retest)**
 - ❖ javítás utáni újra tesztelés (megtalált és javított hiba már nem hiba)
- ❖ **Regressziós tesztelés**
 - ❖ a változtatások okozta "mellékhatások" tekintetében
 - ❖ általában kiterjedt tesztkészlet
 - ❖ szoftverrel párhuzamos karbantartást igényel

Alfa- és bétatesztek

- ❖ COTS szoftverek esetén a megrendelői elfogadási tesztek helyett
 - ❖ szoftvertervezési és kódolási hibák felfedezése
 - ❖ a felhasználói csoport minőségre vonatkozó visszacsatolása
 - ❖ **valódi felhasználási helyzetek**, elfogultság csökkentése, szélesebb tesztelői kör
- ❖ **Alfateszt**
 - ❖ a fejlesztés helyén
- ❖ **Bétateszt**
 - ❖ a potenciális felhasználóknál